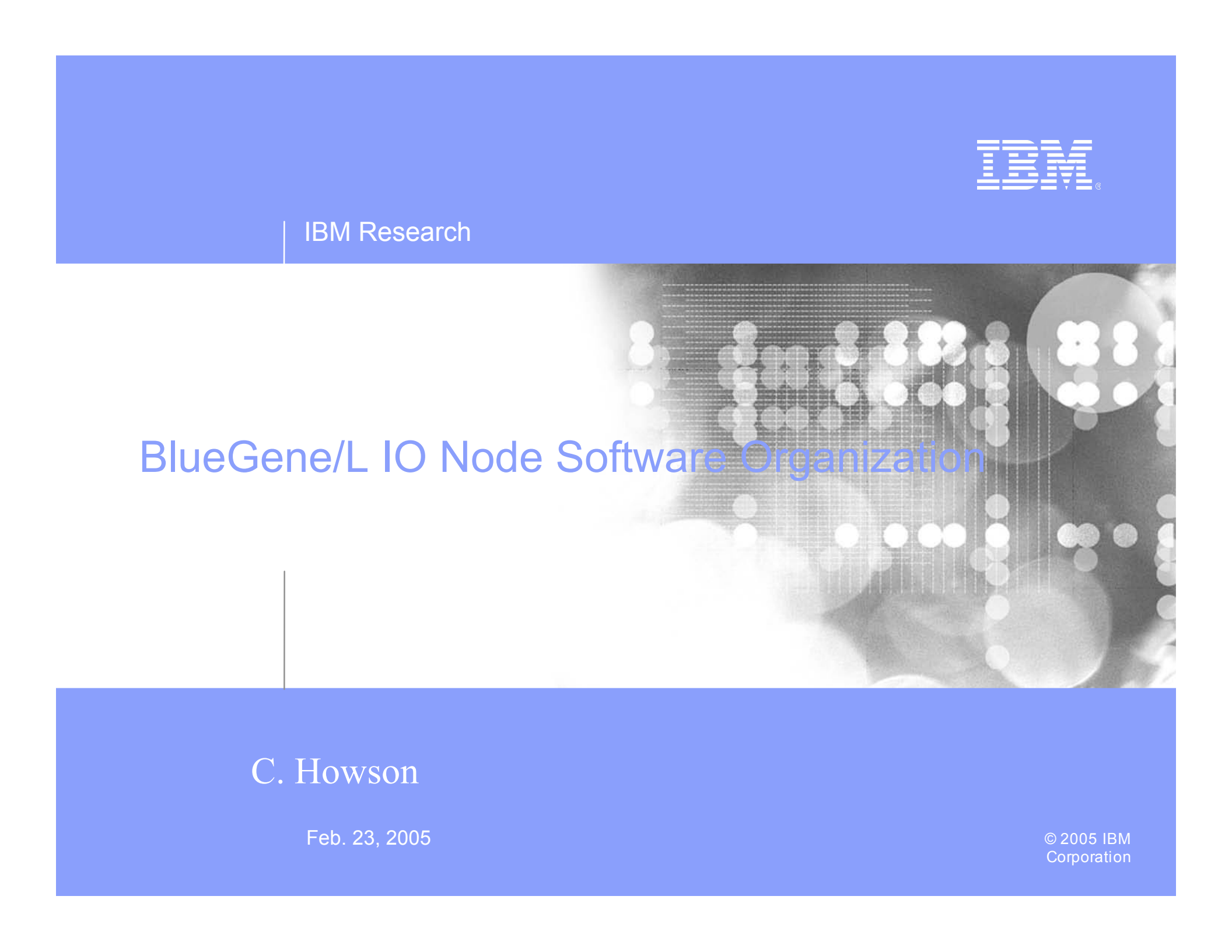


IBM Research

BlueGene/L IO Node Software Organization

C. Howson

Feb. 23, 2005

Outline

- Hardware Configuration
- Software Components on IO node
  - Kernel
  - Ramdisk
  - NFS mounted FS
  - BlueGene/L specific
- Performance
  - Network
  - File IO

IO Node Hardware

- IO nodes are same ASIC as Compute nodes, with different network connections wired
- ASIC has 4 network connections:
  - control, tree, torus, GigE
- Compute node has 3:
  - control, tree, torus
- IO node has 3
  - control, tree, GigE
- GigE network is only network suitable for high bandwidth IO

IO Node Software

- Linux kernel (2.4.19 + patches) on 1 CPU
- Minimal ramdisk based distribution (busybox)
- NFS mounted filesystem provides additional components (/bgl)
- BlueGene/L specific ciod
  - job management
  - system call functions for CNs

Boot Process

- Inital ramdisk and kernel image loaded via control network
- Kernel start message via control network
- Kernel boots
- Init starts
- RC scripts
  - /bgl NFS mount
- Extra kernel modules loaded
- Ciod starts
- Ready to accept jobs

IO node output of 'ps ax' after boot

```
PID TTY      Uid       Size State Command
  1   0          980   S   init
  2   0            0   S   [keventd]
  3   0            0   S   [ksoftirqd_CPU0]
  4   0            0   S   [kswapd]
  5   0            0   S   [bdflush]
  6   0            0   S   [kupdated]
 26   0            0   S   [rpciod]
 68   1         1468   S   /sbin/portmap
 72   0         6620   S   /sbin/ciod.440
 80   0          996   S   /bin/sh
 81   0          988   R   ps ax
```

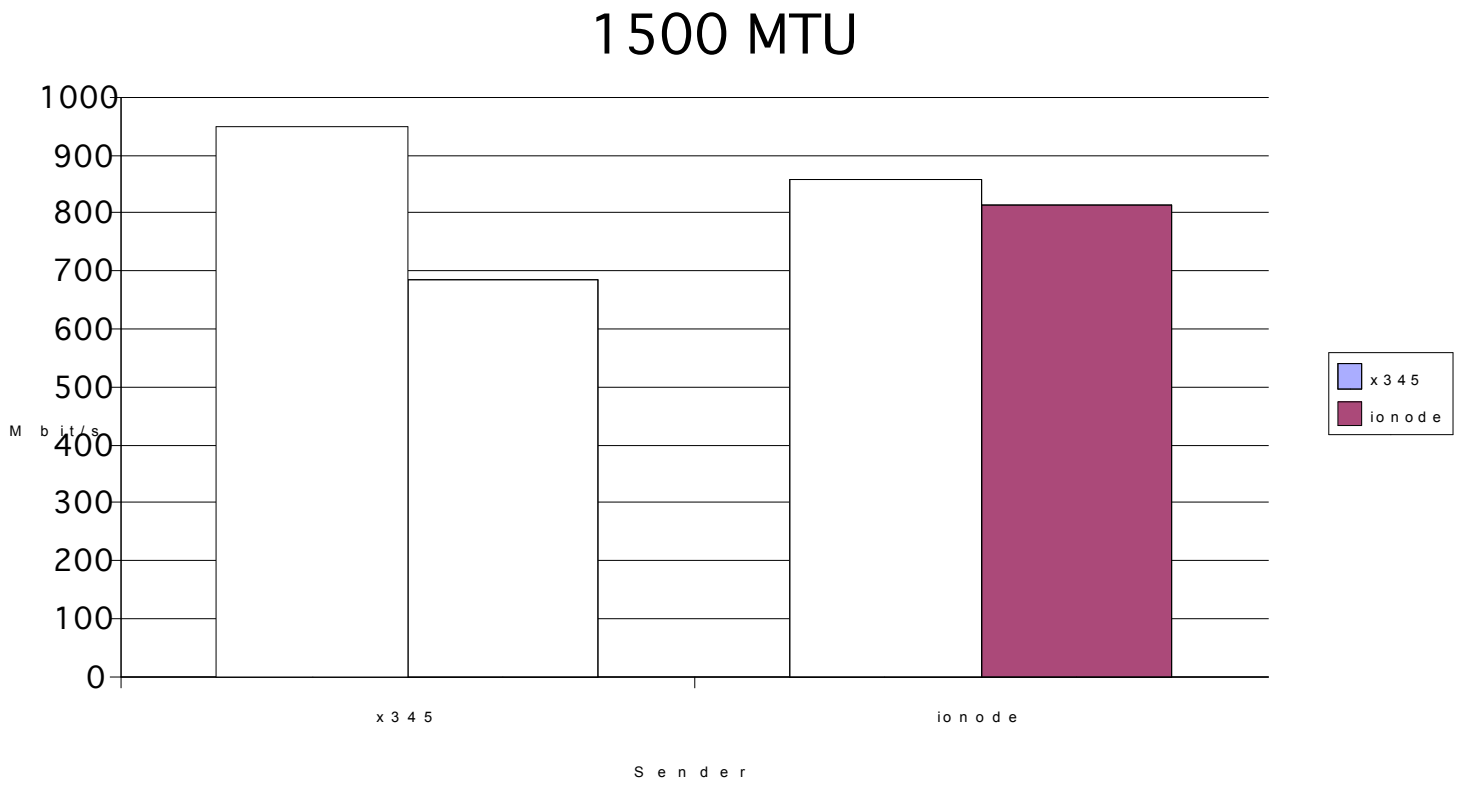
Design Philosophy

- IO node is more like an embedded system than general purpose computer
- Prefer static to dynamic memory allocation
  - CIOD is a single process: don't even fork for a new job
  - CIOD has relatively small fixed size buffers/compute node
- Avoid many daemons
  - consume memory
  - slow down IO due to scheduling conflicts
  - CPU is not fast wrt GigE
- Simple design
  - not fancy, but works
- Site specific configuration possible by modifying rc scripts
  - But try to offload functionality to external servers

Network performance: key to file IO performance

- Factors: MTU, sysctl, switch
- MTU: GigE supports Jumbo frames - up to 9000 vs standard 1500 bytes
  - Every host on physical net must be configured to accept jumbo frames
  - Private network helps for this
- sysctl: Linux default tcp settings are conservative (slow for GigE)
  - Modify apps to customize settings or change defaults
  - Set good defaults: /proc/sys/net/core/{r,w}mem_default
  - I like receive buffers to be 2x send buffers (512k,256k)
  - ifconfig txqueuelen 10000 helps a bit
- Network Switches are not crossbars
  - Keep IO nodes close to File servers on switch
  - Physical wiring choice affects performance

Netperf performance

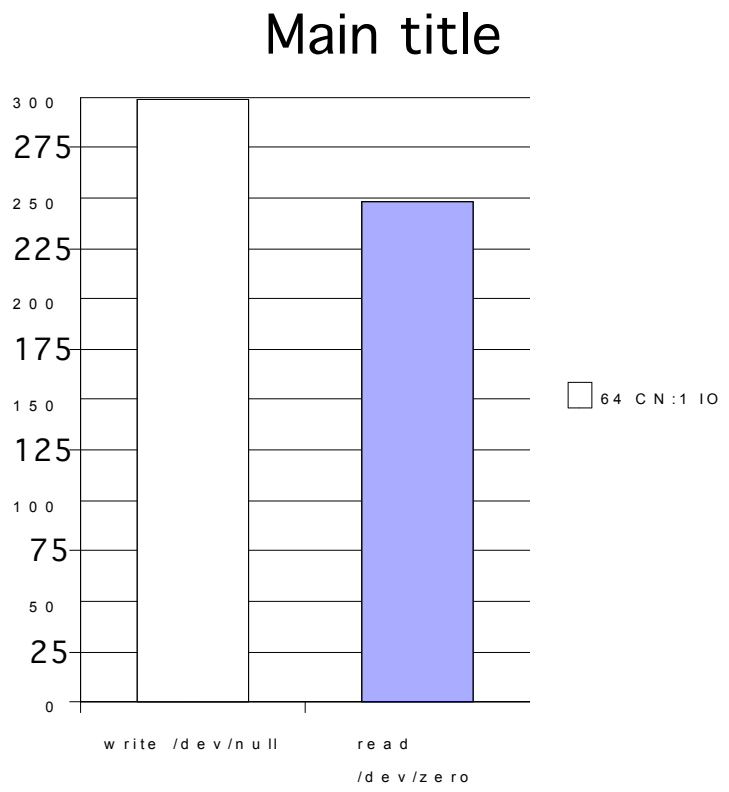


File IO Performance

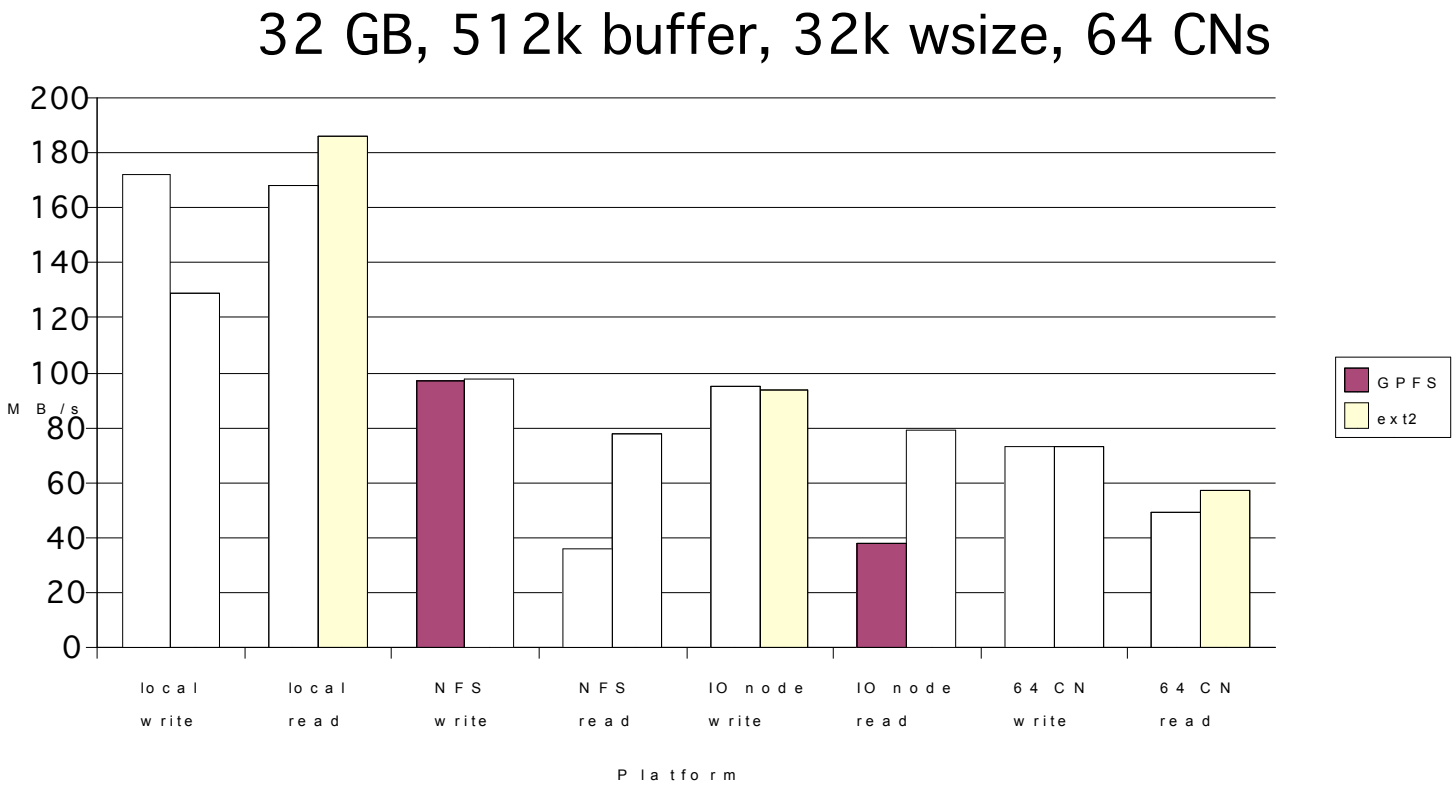
- IO node has very little RAM compared to aggregate compute node ram
  - Not so much opportunity to cache files on IO node
- Primary concern is streaming large files out to file servers
- Bottleneck is ciod/linux-fs/nwk, not CNK/tree/ciod
  - IO node software environment must be light and fast

CNK/tree/CIOD Performance

- Tree max theoretical bandwidth is 350 MB/s
- Measure by writing to /dev/null, reading from /dev/zero
- Read involves kernel zeroing buffers: 50 MB/s slower



GPFS and Ext2 Performance



Conclusion

- IO node software is very simple
- Filesystem and networking overhead is quite high
  - Be careful when adding daemons